International Journal of Grid and Distributed Computing
Vol. 9, No. 1 (2016), pp.193-200
http://dx.doi.org/10.14257/ijgdc.2016.9.1.19

Research on Cloud Resource Section Method for the Multi-layer Ontology

Zhang Hong-lie[1], Li Xin[1], LIU Yan-ju[2], and Li Cheng[1]

[1] *Communication and Electronic Engineering Institute, Qiqihar University, Qiqihar, China*
[2] *Computing Center, Qiqihar University, Qiqihar, China*

**15845673377@163.com*

Abstract

To solve the problem of cloud resource discovery in the interconnected environment for multiple providers, a resource selection method for the multi-layer ontology is propose from the view of software developer requirements. The ontology is adopted to describe the user requirements, the resource constraint requirements and the cloud resource attribute firstly. Then, the bidimensional matching algorithm is constructed based on two principles of concept matching and attribute matching. And the resource discovery and resource selection are realized by the algorithm, combine with the ontology mapping. The method not only implements the resource selection among cloud providers, but also eliminates the implementation details in the process of resource selection. It overcomes the technical limitations of the single cloud provider. The simulation results show that the importance of the resource requirements is adopted by bidimensional matching algorithm to carry out element level matching which improves the recall and precision of resources greatly.

***Keywords**: Cloud resource selection; User requirements; Ontology; Bidimensional matching.*

1. Introduction

The cloud computing can provide on demand dynamically scalable virtualized resources via internet. In fact, the emergence of cloud computing has not only changed the way of providing IT service but influenced the way of application development, which helps organizations to save resources during the whole cycle and shorten application development time. It has become necessary how to customize the most appropriate resource sets for software developers to develop application and carry out execution. The cloud resource selection method has become a key issue.

There have been many researches aiming at the issue of "cloud service search or cloud resource discovery". Kang *et al*. [2] proposed a cloud service agent system based on ontology to realize the cloud service selection; Fortis *et al*. [3] proposed a ontology description method of cloud service and laid a foundation for the implement of simplifying application development and deployment. Liu *et al*.[4] used a novel adaptive resource selection method to solve the possibility of resource sharing among mobile devices. An ontology based on resource discovery mechanism was presented by Vaithiya *et al*.[5], which was used to address the complexity of resource discovery in mobile grid environment. From the researches above, the existing methods have many shortcomings. From the perspective of resource providers, it is difficult for the traditional resource selection methods to be maximized to meet the requirements of application, causing the overflow ability of providing resources to be overflowing. On the other hand, as their own APIs are usually not compatible, different APIs are offered by different providers for

ISSN: 2005-4262 IJGDC

requesting and configuring resources, causing the application of developer being locked for specific provider.

In this paper, a cloud resource selection model based on ontology was proposed from the perspective of software developer' requirements. The bidimensional matching algorithm is constructed on the basis of resource matching hard requirements and elastic requirements. And the most appropriate resource for meeting the requirements is selected out, combined with the ontology mapping model by Sun [9].

The remainder of this paper is organized as follows. In Section 2, the cloud resource organization mode and search mechanism are presented. In Section 3, the multilayer ontological model is described. The bidimensional matching algorithm is put forward in Section 4. Some simulations and experiments are conducted and analyzed in Section 5. Conclusions and future work are summarized in the end.

2. Cloud Resource Organization Mode and Search Mechanism

After the interconnection between cloud, all the cloud resource providers are unified to carry out comprehensive management and form a cloud resource pool. For all kinds of users, the universal API interface is set in cloud resource pool for selecting resources [60-62]. From the overall view, the alternative resources for user are concentrated in an unknowable API interface for provider, and the principle diagram of cloud resource organization mode is shown in Figure 1.

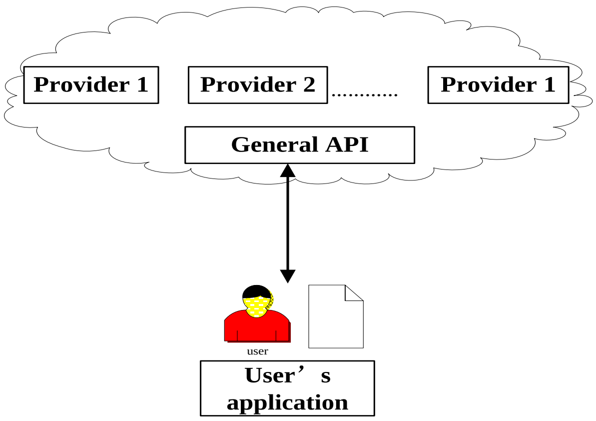

Figure 1. Cloud Resource Organization Mode Principle

The resource users and resource provider are separated by the surface interface obviously, and the excessive intervention and restriction from providers are blocked in resource selection process for users. The mode provides the possibility for considering user individual requirements. The corresponding cloud resource search mechanism in Figure 1 is built. And it consists of user layer, mapping layer and resource layer, which is shown in Figure 2.

For user layer, the user layer is made up of the domain-specific application developers. This layer is divided into the task expression and the resource requirement expression. For mapping layer, resource matching is implemented through user's resource requirements mapped to the underlying cloud resource pool. Considering the user's preference the sets of cloud resources are selected to meet the requirements of developers in maximum. For resource layer, this layer consists of many cloud resource vendors, and different vendors supply different kinds of cloud resources by which the infinite resource pool is formed.

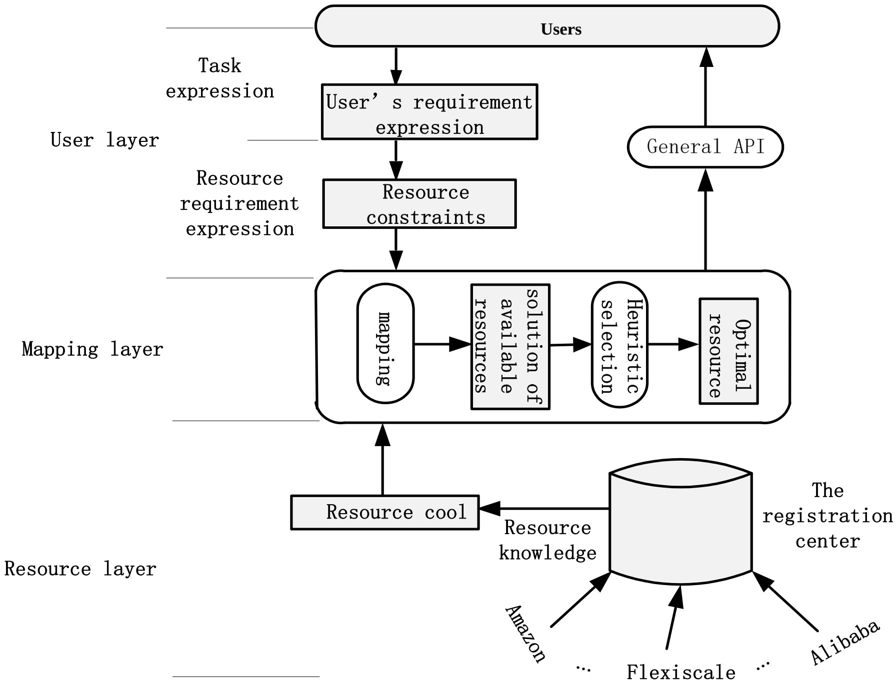

Figure 2. Cloud Resource Discovery Model

3. A Multilayer Ontological Model

According to the proposed cloud resource search method presented in Figure 3, a group of available resources are determined from the viewpoint of software developer. Therefore, ontology should be used to describe the task requirements and resource in Figure 3, which is the effective solution to resource search. In this paper, three ontologies are constructed, namely the task ontology, the resource requirements ontology and the resource ontology.

3.1. Task Ontology

It is used to describe application development requirements. The ontology focuses on user requirements and provides a semantic mechanism for capturing user requirements in a language or terminology familiar with user's application domain. The ontology consists of two sets, which is shown in Expression (1).

Task ontology = {***F-requirement***, ***S-requirement*** } (1)

F-requirement —— a functional requirement set, which describes the needful detail function of the application.
S-requirement —— a service requirement set, which describes some other requirements such as the runtime requirement of the application, besides function description.

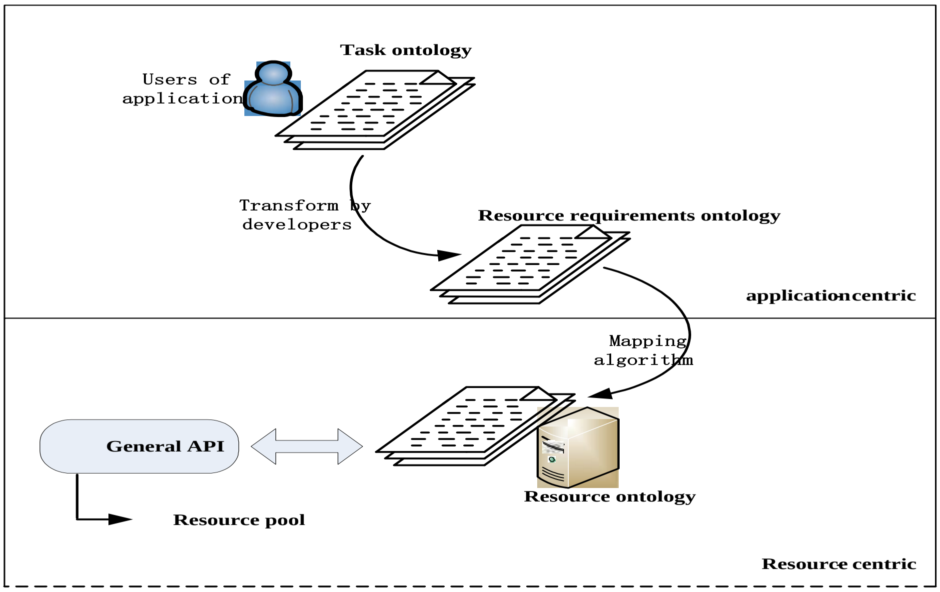

Figure 3. Mapping Relationship among Ontologies

3.2. Resource Requirements Ontology

It is used to describe the generic resource model that satisfies the constraints of the task ontology. Resource requirement ontology is a transformation of the task ontology in user layer, and the transformation process is completed by software developers who refer to the available historical data [11]. The ontology is a precise representation for the task ontology, which is often used to find and select the most appropriate resource group from resource layer. In the proposed model, the resource requirement ontology includes three main classes: ***Infrastructure, Infrastructure Requirement, and Constraints***, which is shown in Expression (2).

Resource requirements ontology = {***Infrastructure Infrastructure Requirement, Constraints***} (2)

3.3. Resource Requirements Ontology

The resource ontology, the bottom layer of the proposed model, defines the properties of the resource provided by different vendors. The methods presented in [4] is more practical, and it describes the cloud resource form the viewpoint of the resource ability. When the multi-layer ontology is defined, if the ontology mapping model proposed by Sun in [65][9]is adopted, the cloud research is turned into the ontology mapping that can be realized by the matching algorithm.

4. Bidimensional Matching Algorithm

According to the definition, resource requirements ontology is a collection that satisfies the task ontology constraint conditions which include hard requirements and elastic requirements. The bidimensional matching algorithm is constructed to realize the mapping among ontologies, according to the classic matching algorithms [9][17].

The resource requirement ontology R and the resource ontology S are expressed as the vector set respectively. The resource requirement ontology is expressed as $\mathrm{R}=\{\mathrm{R}_1, \mathrm{R}_2 \ldots \mathrm{R}_n\}$, where $R_i (\mathrm{i} \le \mathrm{n})$ represents the corresponding concept in ontology, the elements of R_i like $R_i[\mathrm{r}]$ represent various requirements for resources, which is shown in Section 3 carefully. Some of these elements are the hard requirements with a high priority with weighted value of 1, and the remainders are the soft requirements with weighted value of ω_t, where $\omega_t \in [0,1]$, and $\sum_{t=0}^{x} \omega_t = 1$, according to user expectations. Matching algorithm is carried out through two stages.

In the first stage, the hard requirements are used as the first dimensional conditions to match and select out the solution of available resources.

First, detect the weight (ω) of every element from component of vector *R*, matching the requirement that $\omega = 1$ with the corresponding from resource ontology according to the formula (1). The matching includes the concept matching and the attribute matching, and there are thresholds Tc and Tp for these two kinds of matching, which is shown as follows:

$$Sim_h(\mathrm{R}_i, \mathrm{S}_i) = Sim_{hconp}(\mathrm{R}_i, \mathrm{S}_i) + Sim_{hprop}(\mathrm{R}_i, \mathrm{S}_i)(\mathrm{i} = 1, 2 \ldots n) \quad (1)$$

$Sim_{hconp}(\mathrm{R}_i, \mathrm{S}_i)$ is the concept matching, and if $Sim_{hconp}(\mathrm{R}_i, \mathrm{S}_i) \leq Tc$,then match successfully.

$Sim_{hprop}(\mathrm{R}_i, \mathrm{S}_i)$ is the attribute matching. $Sim_{hprop}(\mathrm{R}_i, \mathrm{S}_i)$ is expressed as $Sim_{hprop}(\mu(\mathrm{R}_i), \mu(\mathrm{S}_i), \mathrm{C})$, and the matching accords to the formula(2) to work [9].

$$Sim_{hprop}(\mu(\mathrm{R}_i), \mu(\mathrm{S}_i), \mathrm{C}) = 1 - \left| \frac{\mu(\mathrm{R}_i) - \mu(\mathrm{S}_i)}{C_{\max} - C_{\min}} \right| \quad (2)$$

$\mu(\mathrm{R}_i)$ and $\mu(\mathrm{S}_i)$ respectively represent digital attribute of concepts R_i and S_i. *C* represents concept referred before. And $C_{\max}$ and $C_{\min}$ represent range of maximum and minmum of attribute. If $Sim_{hprop}(\mu(\mathrm{R}_i), \mu(\mathrm{S}_i), \mathrm{C}) \leq T_p$ exists, then match successfully.

Then, according to the result of hard requirement matching $Sim_h(\mathrm{R}_i, \mathrm{S}_i) \leq Tc + Tp$, the available solution is chosen out preliminarily, and the amount of the resources in preliminary solution is *x* that is part of the total resources provided by the resource pool. All the resources contained in the preliminary solution satisfy the functional requirements of the application developers.

In the second stage, the soft requirements are used as the second dimensional conditions to match and select the most appropriate resources, which is as followed:

Match every soft requirement of every component of vector with the concept contained in the preliminary solution S_x according to the formula (1) to obtain $Sim_s(\mathrm{R}_i, \mathrm{S}_i)(\mathrm{i} \leq n)$, which represents the value of soft requirement matching. The matching of the soft requirements contains the concept matching and the attribute matching like the first stage. The two kinds of matching are also respectively through the calculation of formula (1) and (2), then $Sim_{sconp}(R_i, S_i)$ and $Sim_{sprop}(R_i, S_i)$ are obtained. According to the formula (3), the matching value $Sim_s(\mathrm{R}, \mathrm{S})$ between vector *R* and preliminary solution S_x is calculated, which is expressed as:

$$Sim_s(\mathrm{R}, \mathrm{S}) = \sum_{t=1}^{x} \omega_t Sim_s(\mathrm{R}_t, \mathrm{S}_t) \quad (3)$$

where $\omega_t \in [0,1]$ is the effect degree of every soft requirement. A higher weight is given by developers to the user's most desired requirement. Finally, according to the soft requirements, the preliminary solution is filtered by matching. For example, if the cost requirement is defined as the most important requirement, it was given a higher weight according to the method proposed, which helps developers to find out the minimum cost of the available resources.

5. Simulation and Experiment

In order to verify effectiveness of the bidimensional matching algorithm cloud resource discovery model, a series of experiments are carried out. Traditional keywords based on resource discovery methods are compared with the proposed method. Three indicators are used to measure the two research methods.

Search time refers to the time that is taken to find and match the most appropriate resources.

Recall ratio refers to the percentage of number of related resources that are searched out in the related resources which are contained in the resource pool.

Precision ratio refers to the percentage of number of suit resource in the amount of relevant resources in the pool.

Now there is development project, and the resource requirements are as follow:

Operating system: Ubuntu Maverickc(hard requirement 1).

Computing resource: 1 CPU, 1GB RAM, and 20GB hard disk space(hard requirement 2).

The resource must be located in China(hard requirement 3).

Maximum budget for the resource is 3 yuan per hour for 30 days(soft requirement).

Formulate a set contained 500 test dates to simulate the cloud resource pool which is composed by the federated multi-providers, store the test dates in the MYSQL as the UDDI, then use software protégé to model the ontology, and the Pellet reasoning machine to traverse all the concepts and attributes in the ontology repository. Construct the ontology according to the development task requirement.

A series of simulations are carried out by comparing the result of using the bidimensional matching algorithm model and that of using the key word based on discovery method, the number of the request resource is varies among1-10, and the experimental results are shown in Figures 4-6.

From Figure 4, it can be seen that the key word based on discovery method smartly reduces the search time compared with the bidimensional matching algorithm model when the same number of target resources are found. That is because the time of the matching calculation is saved for concepts and attributes of the ontology, which leads to the search time and number of target resource in a linear relationship.

From Figure 5, it can be seen that fewer resources can be found through the key word search. Because the ontology is used to describe resource requirements and resource, the understanding ability of computer is increased and the possibility of the most suitable match is added. So the bidimensional matching algorithm resource discovery model is adopted to find more related resources.

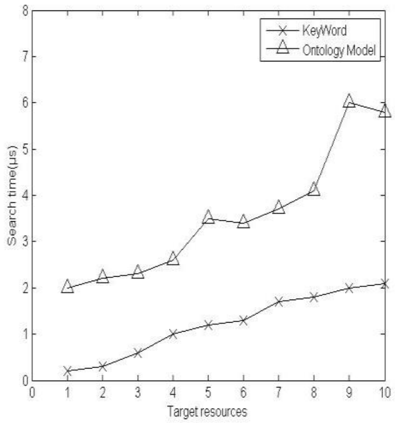

Figure 4. Search Time Comparison

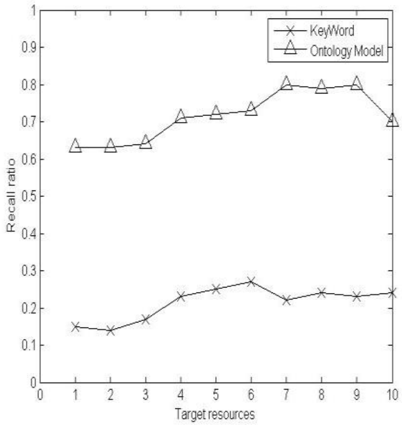

Figure 5. Recall Ratio Comparison

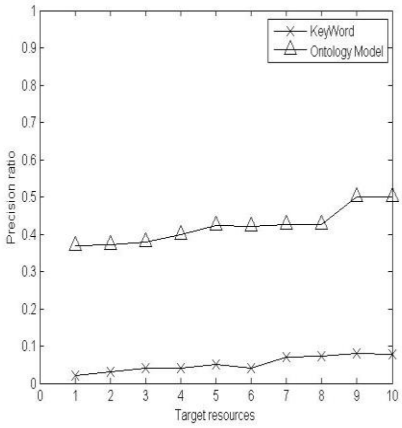

Figure 6. Precision Ratio Comparison

From Figure 6, it can be seen that only by key word research, the most appropriate resource matching precision ratio is close to zero. In addition to the used semantic, the bidimensional matching algorithm makes precision higher by filtering resource subtly.

Conclusions

Based on the multi-provider interconnection background and the cloud resource organization mode, from user requirements of software development, a cloud resource discovery and selection method is presented based on multi-layer ontology and the cloud resource search mechanism to obtain the cloud resources that satisfies the requirements in this paper. On one hand, the ontology is used as the development tool, the underlying useful information is mined by knowledge reasoning, which makes computer understand users' expression information and improves the precision ratio of resource discovery. On the other hand, the semantic comprehension of resource discovery process is strengthened by the coherence of resource description and resource requirements description, which improves the recall ratio of resource inquiry. The bidimental matching algorithm is presented to carry out matching in phrase, which improves the efficiency of resource selection for specific demand in the market. A theoretical basis and feasible method is introduced for the flexible use of vast amounts of cloud resources by this research.

Acknowledgment

This work was jointly supported by the National Natural Science Foundation of China (Grant No. 61403222), and Natural Science Foundation of Heilongjiang Province (Grant No. F201439, F201440).

References

[1] D. Bernstein and D. Vij, "Intercloud directory and exchange protocol detail using XMPP and RDF", Proceedings of the 6th World Congress on Services (SER-VICES-1), Miami, **(2010)**, pp. 431-438.
[2] J. Y. Kang and M. S. Kwang, "Towards agent and ontology for cloud Service discovery", International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, **(2011)**, pp. 483-490.
[3] T. F. Fortis, V. I. Munteanu and V. Negru, "Towards an ontology for cloud service". Sixth International Conference on Complex, Intelligent, and Software Intensive Systems, **(2012)**, pp. 787-792.
[4] W. Liu, T. Nishio, R. Shinkuma and T. Takahashi, "Adaptive resource discovery in mobile cloud computing", Computer Communications, vol. 50, **(2014)**, pp. 119-129.
[5] V. Stephen and M. S. Bhanu, "Ontology based resource discovery mechanism for mobile grid environment", 2013 Second International Conference on Advanced Computing, Networking and Security, **(2013)**, pp. 154-159.
[6] L. Kapoor, S. Bawa and A. Gupta, "Hierarchical chord-based resource discovery in intercloud environment", IEEE/ACM 6th International Conference on Utility and Cloud Computing, **(2013)**, pp. 464-469.
[7] W. Q. Li, X. Sun, C. Y. Zhang and Y. Feng, "A Semantic Similarity Measure between Ontological Concepts", Acta Automatica Sinica, vol. 38, no. 2, **(2012)**, pp. 229-235.
[8] W. J. Liu, N. Guo and T. G. Jin. Similarity calculating model based on manufacturing resource ontology", Computer Integrated Manufacturing Systems, vol. 16, no. 20, **(2010)**, pp. 2468-2474.
[9] L. Kang, G. S. Chen, S. L. Wang, Q. Li, L. Guo and W. Y. Song, "Ontology based process resources discovery for cloud manufacturing", Computer Integrated Manufacturing Systems, vol. 19, no. 9,**(2013)**, pp. 2325-2331.
[10] R. Bennyr, D. Breitgand, E. Levy, *et al*, "The reservoir model and architecture for open federated cloud computing [J]", IBM Journal of Research and Development, **(2009)**, pp. 53-76.
[11] OASIS UDDI Specification TC, at http://www.oasisopen.org/committees/tc_home.php?wg_abbrev=uddispec.
[12] T. Harmer, P. Wright, C. Cunningham, J. Hawkins and R. Perrott, "An application-centric model for cloud management[C]", Proceedings of the 6th World Congress on Services (SERVICES-1), **(2010),** pp.439-446.

